Causal Discovery with Fewer Conditional Independence Tests

Kirankumar Shiragur [*][1] Jiaqi Zhang [*][1][2] Caroline Uhler [2]

Abstract

Many questions in science center around the fundamental problem of understanding causal relationships. However, most constraint-based causal discovery algorithms, including the well-celebrated PC algorithm, often incur an *exponential* number of conditional independence (CI) tests, posing limitations in various applications. Addressing this, our work focuses on characterizing what can be learned about the underlying causal graph with a reduced number of CI tests. We show that it is possible to a learn a coarser representation of the hidden causal graph with a *polynomial* number of tests. This coarser representation, named Causal Consistent Partition Graph (CCPG), comprises of a partition of the vertices and a directed graph defined over its components. CCPG satisfies consistency of orientations and additional constraints which favor finer partitions. Furthermore, it reduces to the underlying causal graph when the causal graph is identifiable. As a consequence, our results offer the first efficient algorithm for recovering the true causal graph with a polynomial number of tests, in special cases where the causal graph is fully identifiable through observational data and potentially additional interventions.

1. Introduction

Causal discovery is a fundamental task in various scientific disciplines including biology, economics, and sociology (King et al., 2004; Cho et al., 2016; Tian, 2016; Sverchkov & Craven, 2017; Rotmensch et al., 2017; Pingault et al., 2018; de Campos et al., 2019; Reichenbach, 1956; Woodward, 2005; Eberhardt & Scheines, 2007; Hoover, 1990; Friedman et al., 2000; Robins et al., 2000; Spirtes et al., 2000; Pearl, 2003). Directed acyclic graphs (DAGs) stand out as a popular choice for representing causal relations, with edge directions signifying the flow of information between variables. The core objective of causal discovery is to identify both the edges and their orientations based on available data. While certain structures can be recovered from observational data (Verma & Pearl, 1990), orienting the full graph often requires additional experiments or interventions.

Research on causal structure learning from observational data dates back to the 1990s (Verma & Pearl, 1990; Spirtes et al., 1989). As a pioneering work in this direction, the PC algorithm (Spirtes et al., 2000), named after the authors Peter Spirtes and Clark Glymour, still remains one of most popular and widely used algorithms. It recovers the structure using observational data through conditional independence (CI) tests, with the number of tests being exponential in the degree of the graph. Following this, many causal discovery algorithms emerged (Kalisch & Bühlman, 2007; Brenner & Sontag, 2013; Alonso-Barba et al., 2013; Schulte et al., 2010), accommodating diverse and more general settings, including the presence of latent variables (Spirtes et al., 1999; 2013) and interventional data (Eberhardt et al., 2005; 2006). However, a common challenge shared by these algorithms is their reliance, to different extents, on an *exponential* number of CI tests in certain graph parameters. This inherent dependence on an exponential number of tests poses practical challenges, making them unsuitable for many real-world scenarios. Moreover, it suggests that achieving *exact* causal structure learning can be highly challenging.

As performing exponential number of tests is limited in many applications, it motivates us to study the following question:

What useful information about the underlying causal graph can be inferred with fewer conditional independence tests?

Aligned with this motivation, our work also explores the role of interventions in the structure learning process.

In our work, we study these questions under standard Markov, faithfulness and causal sufficiency assumptions (Lauritzen, 1996; Spirtes et al., 2000). The primary contribution of our work is an efficient algorithm that uses a *polynomial* number of CI tests and recovers a representation

[*]Equal contribution; alphabetic order [1]Eric and Wendy Schmidt Center, Broad Institute [2]Laboratory for Information & Decision Systems, Massachusetts Institute of Technology. Correspondence to: Kirankumar Shiragur <kshiragu@broadinstitute.org>, Jiaqi Zhang <viczhang@mit.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

of the underlying causal graph with observational and optionally interventional datasets. This representation consists of a partition of the vertices and a DAG defined over its components which is consistent with the underlying causal graph. In addition, our representation is designed to avoid dummy partitions that group all the vertices into a single component. The definition of our representation ensures that the components in the partition satisfy several additional properties, guaranteeing that each component either contains a single vertex or comprises an edge that could only be oriented after an intervention is performed on one of its endpoints. We refer to this representation as the Causally Consistent Partition Graph (CCPG) representation.

An important implication of our results is that if the underlying causal graph is fully identifiable using only observational data, our algorithm yields a CCPG with a partition containing components, each of which is of size one. The size-one property of each component means that our algorithm recovers the true causal graph using only a *polynomial* number of conditional independence tests. We extend this result in the presence of interventions and provide an algorithm that recovers the true causal graph, when the set of interventions provided is sufficient to identify the underlying causal graph. To the best of our knowledge, our algorithms present the first to guarantee recovering the true causal graph using a polynomial number of tests when the graph is either entirely identifiable from observational data or with an additional set of interventions.

1.1. Related Works

Efficient algorithms for causal structure learning (Spirtes et al., 2000; Claassen et al., 2013) exist for constant bounded degree graphs, recovering the causal graph with a polynomial number of CI tests. For general causal graphs, current methods often entail an exponential number of CI tests, where (Xie & Geng, 2008; Zhang et al., 2024) aimed to to reduce such complexity. For Bayesian network learning, finding a minimal Bayesian network is NP-hard, even with a constant-time CI oracle and nodes with at most $k \geq 3$ parents. Chickering et al. (2004) demonstrated this hardness through a polynomial reduction from the NP-complete problem, Degree-Bounded Feedback Arc Set. These findings highlight the contrast between causal structure and minimal Bayesian network learning, suggesting that causal structure learning is notably more straightforward. Our results further reinforce this notion by identifying a special class of causal graphs that can be recovered with a polynomial number of conditional independence tests. For other hardness results on Bayesian network learning, we refer readers to Bouckaert (1994); Chickering et al. (2004) and references therein.

Learning causal relationships from observational data (Verma & Pearl, 1990; Spirtes et al., 1989; 2000; Chickering, 2002; Geiger & Heckerman, 2002; Nandy et al., 2018) and interventional data (Eberhardt, 2010; Hu et al., 2014; Shanmugam et al., 2015; Greenewald et al., 2019; Squires et al., 2020; Choo et al., 2022; Choo & Shiragur, 2023; Shiragur et al., 2024) is a well studied problem with a rich literature. We encourage interested readers to explore Glymour et al. (2019); Squires & Uhler (2022) and references therein for a more comprehensive understanding and further details.

1.2. Organization

The rest of our paper is organized as follows. Section 2 is our preliminary section. In Section 3, we provide all the main results of the paper. In Section 4 and Section 5 combined, we provide our CCPG recovery algorithm when just observational data is available. In Section 6, we extend our results to the case of interventions. We provide numerical results in Section 7. Finally in Section 8, we conclude with a short discussion and few open directions.

2. Preliminaries

2.1. Graph Definitions

Let $\mathcal{G}$ be a directed acyclic graph (DAG) on n vertices in V. For a vertex $v \in V$, let $\texttt{Pa}(v), \texttt{Anc}(v)$, and $\texttt{Des}(v)$ denote the *parents*, *ancestors*, and *descendants* of v respectively. Let $\texttt{Anc}[v] = \texttt{Anc}(v) \cup \{v\}$ and $\texttt{Des}[v] = \texttt{Des}(v) \cup \{v\}$.

For a set of vertices $S \subset V$, denote $\texttt{Pa}(S) = \cup_{v \in S} \texttt{Pa}(s)$. Similarly define $\texttt{Anc}(S), \texttt{Des}(S), \texttt{Anc}[S]$, and $\texttt{Des}[S]$. We write $\mathrm{src}(S)$ as the set of *source* nodes within S, that is,

$$\mathrm{src}(S) = \{v \in S \mid \texttt{Anc}(v) \cap S = \varnothing\} .$$

Denote $\bar{S} = V \backslash S$. Let $\mathcal{G}[S]$ be a *subgraph* of $\mathcal{G}$ by removing all vertices in $\bar{S}$.

A *v-structure* refers to three distinct vertices u, v, w such that $u \to v \leftarrow w$ and u, w are not adjacent. An edge $u \to v$ is a *covered edge* (Chickering, 1995) if $\texttt{Pa}[u] = \texttt{Pa}(v)$. A *path* in $\mathcal{G}$ is a list of distinct vertices, where consecutive vertices are adjacent. We can associate a *topological ordering* $\pi : V \to [n]$ to any DAG $\mathcal{G}$ such that any $u \to v$ in $\mathcal{G}$ satisfy $\pi(u) < \pi(v)$. Note that such topological ordering is not necessarily unique.

2.2. D-Separation and Conditional Independence

DAGs are commonly used in causality (Pearl, 2009), where vertices represent random variables and their joint distribution P factorizes according to the DAG: $P(v_1, \ldots, v_n) = \prod_{i=1}^{n} P(v_i \mid \texttt{Pa}(v_i))$. This factorization entails a set of conditional independencies (CIs) in the *observational* distribution P. These CI relations are fully characterized by *d-separation* (Geiger & Pearl, 1990). Formally, for disjoint vertex sets $A, B, C \subset V$, sets A, B are *d-separated* by C

if and only if any path connecting A and B in $\mathcal{G}$ is inactive given C. A path is *inactive* given C when it has a *collider*[1] $d \notin \mathtt{Anc}[C]$ or a non-collider $c \in C$; otherwise the path is *active* given C. Figure 1 illustrates these concepts.

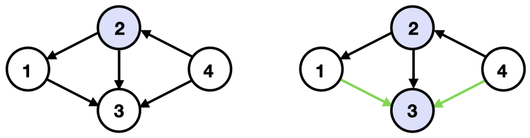

Figure 1. **(Left).** $\{1\}$ and $\{4\}$ are *d-separated* by $\{2\}$, as all paths are *inactive* given $\{2\}$. **(Right).** $\{1\}$ and $\{4\}$ are *not* d-separated by $\{2, 3\}$, as path $1 \to 3 \leftarrow 4$ is *active* given $\{2, 3\}$ by *collider* 3.

We write $A \perp B \mid C$[2] when A, B are conditionally independent given C in the observational distribution P. If any set among A, B, C contains only one node, e.g., $A = \{a\}$, we write $a \perp B \mid C$ for simplicity. When C d-separates A, B, then it holds that $A \perp B \mid C$ (known as the global Markov property (Geiger & Pearl, 1990)). Under the faithfulness assumption, the reverse also holds, i.e., all CI relations in P are implied by d-separation in $\mathcal{G}$.

Setup. In this work, we assume that the causal DAG $\mathcal{G}$ is *unknown*. But we assume causal sufficiency (i.e., no latent confounders), faithfulness and access to enough samples from P to determine if $A \perp B \mid C$ for any $A, B, C \subset V$. As all CIs are implied by d-separations, we may infer information about $\mathcal{G}$ using these tests.

2.3. Interventions

An *intervention* $I \subset V$ is an experiment where the conditional distributions $P(v \mid \mathtt{Pa}(v))$ for $v \in I$ are changed into $P^I(v)$.[3] Such interventions eliminate the dependency between v and $\mathtt{Pa}(v)$. Let $\mathcal{G}^I$ denote the modified version of $\mathcal{G}$, where all incoming edges to $v \in I$ are removed. Let P^I denote the interventional distribution, i.e., $P^I(v_1, \ldots, v_n) = \prod_{v \in I} P^I(v) \prod_{v \notin I} P(v \mid \mathtt{Pa}(v))$. Then P^I factorizes with respect to $\mathcal{G}^I$. We denote $A \perp_I B \mid C$ for CI tests in the interventional distribution P^I.

Setup with Interventions. Similar to the observational setting, we assume faithfulness of P^I to $\mathcal{G}^I$ and access to enough samples from P^I to determine if $A \perp_I B \mid C$.

2.4. Verifying Intervention Sets and Covered Edges

When it is possible to perform *any number of CI tests*: with observational data, a DAG $\mathcal{G}$ is in general only identifiable up to its skeleton, v-structures (Andersson et al., 1997), and possibly additional edges given by the Meek rules (Meek, 1995). Identifiability can be improved with interventional data (Hauser & Bühlmann, 2012), where I allows us to infer the edge orientation of any edge cut by I and $V \setminus I$.

A *verifying intervention set* $\mathcal{I}$ for a DAG $\mathcal{G}$ (Choo et al., 2022) is a set of interventions that fully orients $\mathcal{G}$, possibly with repeated applications of the Meek rules. We will make use of the following result in our work.

Proposition 2.1 (Theorem 9 in (Choo et al., 2022))**.** *Set $\mathcal{I}$ is a verifying intervention set if and only if for every covered edge $u \to v$ in $\mathcal{G}$, there is $|I \cap \{u, v\}| = 1$ for some $I \in \mathcal{I}$.*

The *verification number* $\nu(\mathcal{G})$ is defined as the minimum size of any verifying intervention set of $\mathcal{G}$. This proposition tells us that $\nu(\mathcal{G})$ equals to the minimum size of any vertex cover of the covered edges in $\mathcal{G}$.

3. Main Results

Here we present our main findings. As highlighted in the introduction, the key contribution of our work lies in recovering a representation of the underlying causal graph that satisfies various desirable properties with very few CI tests. We now provide a formal definition of this representation.

Definition 3.1 (CCPG & $\mathcal{I}$-CCPG)**.** A *Causally Consistent Partition Graph (CCPG)* representation of a DAG $\mathcal{G}$ on V consists of a partition of V into components $V_1, \ldots, V_k$ and a DAG $\mathcal{D}$ between the components such that,
(intra-component property): for each $i \in [k]$, it holds that $|\mathrm{src}(V_i)| = 1$. Furthermore, if $|V_i| > 1$, then $\mathcal{G}[V_i]$ has at least one covered edge.
(inter-component property): $\mathcal{D}$ is topologically ordered, i.e., $i \to j$ in $\mathcal{D}$ only if $V_i < V_j$. It is also consistent with $\mathcal{G}$: (1) if there is no directed edge $i \to j$ in $\mathcal{D}$, then there are no edges between V_i and V_j in $\mathcal{G}$; (2) if there is a directed edge $i \to j$ in $\mathcal{D}$, then there is $u \in V_i$ such that $u \in \mathtt{Pa}(\mathrm{src}(V_j))$.

We further define an *Interventional Causally Consistent Partition Graph ($\mathcal{I}$-CCPG)* representation of $\mathcal{G}$ with respect to an intervention set $\mathcal{I}$: an $\mathcal{I}$-CCPG is a CCPG representation of $\mathcal{G}$ that additionally satisfies the following **strong intra-component condition**: for each $i \in [k]$, if $|V_i| > 1$ then $\mathcal{G}[V_i]$ has at least one *unintervened*[4] covered edge.

Figure 2 illustrates these concepts. Note that when $\mathcal{I} = \varnothing$, $\mathcal{I}$-CCPG reduces to CCPG.

In Definition 3.1, the first property prefers finer partitions, while the second property ensures consistency. Formally, we can show the following properties of these representations, which establish the significance of CCPGs. Proofs for all lemmas in this section can be found in Appendix A.

[1] Vertex d is a collider on a path iff $\cdot \to d \leftarrow \cdot$ on the path.

[2] For simplicity, we also write $A \perp B \mid C$ for potential overlapping sets to denote $A \perp B \mid C \setminus (A \cup B)$

[3] We consider hard interventions in this work.

[4] An edge is intervened by I if only one of the vertices is in I.

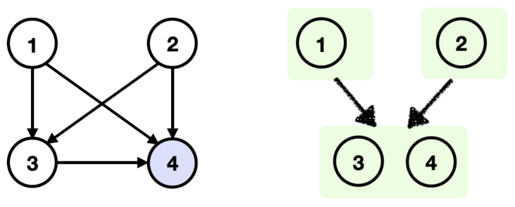

Figure 2. **Example of CCPG & $\mathcal{I}$-CCPG. (Left).** Ground-truth $\mathcal{G}$. **(Right).** A CCPG representation of $\mathcal{G}$, where V_1, V_2, V_3 are indicated by green boxes and $\mathcal{D}$ is illustrated in chalk strokes. Vertices $3, 4$ can be in one component as $3 \to 4$ is a covered edge. For $\mathcal{I} = \{4\}$, the only $\mathcal{I}$-CCPG is $\mathcal{G}$ itself (due to strong intra-component condition in Definition 3.1).

Lemma 3.2 (Properties of CCPG). *For any intervention set $\mathcal{I}$ (including $\varnothing$), the following arguments hold:*

- *$\mathcal{D} = \mathcal{G}$ (i.e., partitioning V into individual vertices) is a valid $\mathcal{I}$-CCPG of $\mathcal{G}$.*
- *If the verification number of $\mathcal{G}$ is zero, i.e., $\nu(\mathcal{G}) = 0$, then $\mathcal{D} = \mathcal{G}$ is the unique valid $\mathcal{I}$-CCPG of $\mathcal{G}$.*
- *If $\mathcal{I}$ is a verifying intervention set of $\mathcal{G}$, then $\mathcal{D} = \mathcal{G}$ is the unique valid $\mathcal{I}$-CCPG of $\mathcal{G}$.*

The key algorithmic contribution of our work, proven in Section 5, lies in an efficient algorithm that learns a valid CCPG with only *polynomial* number of CI tests.

Theorem 3.3 (Learning CCPG). *Given observational data, there exists an efficient algorithm that performs at most $\mathcal{O}(n^5)$ CI tests, and outputs a CCPG representation.*

This result extends to the interventional setting as follows; the proof is given in Section 6.

Theorem 3.4 ($\mathcal{I}$-Learning CCPG). *Given observational data and interventional data from interventions in $\mathcal{I}$, there exists an efficient algorithm that performs at most $\mathcal{O}(n^5) + |\mathcal{I}| \cdot \mathcal{O}(n^3)$ CI tests, and outputs an $\mathcal{I}$-CCPG representation.*

Combining these results with the properties of CCPG in Lemma 3.2, this provides an efficient algorithm for learning the causal graph with polynomial number of conditional independence tests under certain cases, detailed below.

Corollary 3.5 (Causal Discovery with Polynomial CI Tests). *For a DAG $\mathcal{G}$ and its verifying intervention set $\mathcal{I}$ (can be $\varnothing$), our algorithm recovers the full causal graph with at most $\mathcal{O}(n^5) + |\mathcal{I}| \cdot \mathcal{O}(n^3)$ CI tests.*

We remark here that Eberhardt et al. (2012) provides a construction of verifying intervention set of size $\log_2(n)$ that is independent of the underlying DAG $\mathcal{G}$. Together with Corollary 3.5, this implies that our algorithm can learn the full causal graph with at most $\mathcal{O}(n^5)$ CI tests.

To the best of our knowledge, our results present the first formal characterization of the information recoverable about general causal graphs using polynomial number of CI tests. Prior works showed that it is possible to learn *sparse* causal graphs with $n^{\mathcal{O}(k)}$ CI tests (Claassen et al., 2013; Spirtes et al., 2000). Here k is an upper bound on the vertex degrees. Note that, the number of CI tests our algorithm requires, $\mathcal{O}(n^5)$, is a polynomial of n that is independent of any graph parameters.

3.1. Proxy V-structure and Meek Rule Statements

In our derivations, we will make use of the following results, which we believe is of separate interest well beyond the scope of this work.

Lemma 3.6 (Proxy V-Structure). *Let $S \subseteq V$ and $u, v, z \in V \backslash S$. If $u \perp v | S$ and $u \not\perp v | S \cup \{z\}$, then $u, v \notin \texttt{Des}[z]$.*[5]

Definition 3.7 (Prefix Vertex Set). We call $S \subseteq V$ a prefix vertex set if it satisfies: for all $w \in S$, $\texttt{Anc}[w] \cap \bar{S} = \varnothing$ (vertices in S appear first in the topological order).

Lemma 3.8 (Proxy Meek Rule 1). *Let S be a prefix subset. If u, v and w are such that $u \in S$, $v, w \notin S$ and $u \not\perp v | S$ and $u \perp w | S \cup \{v\}$ then $v \notin \texttt{Des}[w]$.*

The results stated above serve as proxy statements of v-structure and Meek Rule 1. Given the CI tests in the preceding lemmas, if we additionally have confirmed adjacencies between specific pairs of vertices, stronger statements could be made; e.g.,in Lemma 3.6, we could conclude that z is a child (or descendant) of both u and v, and in Lemma 3.8 that w is a child of v. However, since our lemma statements do not assume any knowledge of adjacency, we can only ascertain weaker statements, in the first case that u and v are not descendants of z, and in the second case that v is not a descendant of w.

While our proxy results reveal weaker relationships among variables, they achieve this using a constant number of CI tests. Uncovering stronger relationships requires adjacency information, which may entail an exponential number of CI tests. Our main contribution lies in leveraging these weaker relationships, along with other favorable properties embedded in our algorithm, to implement the prefix vertex procedure and reconstruct the CCPG representation of the underlying causal graph using few CI tests.

4. Prefix Vertex Set

The core component of our CCPG algorithm involves a procedure that produces a series of prefix vertex sets. We now show how such prefix vertex sets can be learned by performing few CI tests.

[5]Similar argument is also proven in Lemma 1 of Magliacane et al. (2016).

This will be helpful to obtain a CCPG representation of $\mathcal{G}$, since the components $V_1, \dots, V_k$ satisfy that $V_1 \cup \dots \cup V_i$ is a prefix vertex set for all $i \in [k]$.

4.1. Algorithm for Learning

We begin by presenting our algorithm for learning a prefix vertex set. This algorithm takes as input a prefix vertex set $S \subsetneq V$ (which can be $\varnothing$) and produces a larger prefix vertex set S'.

The analysis of Algorithm 1 will be provided in the next section. For an input prefix vertex set $S \subsetneq V$, it makes use of three types of CI tests, which we formalize below.[6]

Definition 4.1 (Type-I Set D_S)**.** For all $w \in \bar{S}$, let $w \in D_S$ if and only if $u \perp v \mid S$ and $u \not\perp v \mid S \cup \{w\}$ for some $u \in V$ and $v \in \bar{S}$.

By the proxy v-structure in Lemma 3.6, these two CI tests indicate that $v \notin \texttt{Des}[w]$. Therefore w can potentially be a descendant of v. Thus D_S contains vertices that are potential descendants of some other vertex in $\bar{S}$. We will rule out this set when searching for prefix $S' \supsetneq S$. Similarly, we can define a type-II set E_S.

Definition 4.2 (Type-II Set E_S)**.** For all $w \in \bar{S} \setminus D_S$, let $w \in E_S$ if and only if $u \perp v' \mid S \cup \{v\}$ and $u \not\perp v' \mid S \cup \{v, w\}$ for some $u \in S$ and $v, v' \in \bar{S} \setminus D_S$.

We also exclude any type-III set F_S, defined as follows.

Definition 4.3 (Type-III Set F_S)**.** For all $w \in \bar{S} \setminus D_S$, let $w \in F_S$ if and only if $u \not\perp v \mid S$, $u \perp w \mid S \cup \{v\}$, and $v \not\perp w \mid S$ for some $u \in S$ and $v \in \bar{S} \setminus D_S$.

By the proxy Meek Rule 1 in Lemma 3.8, the first two CI tests $u \not\perp v \mid S$ and $u \perp w \mid S \cup \{v\}$ guarantee that $v \notin \texttt{Des}[w]$. The remaining CI test $v \not\perp w \mid S$ is to ensure that we do not exclude too many vertices in F_S, in particular $\text{src}(\bar{S})$, as we show in the next section.

Algorithm 1 Learning a Prefix Vertex Set
```
1: Input: A prefix vertex set $S \subsetneq V$. CI queries from $\mathcal{G}$.
2: Output: A prefix vertex set $S'$ such that $S' \supsetneq S$.
3: Compute type-I set $D_S$.
4: Compute type-II and III sets $E_S, F_S$.
5: Let $U = \bar{S} \setminus (D_S \cup E_S \cup F_S)$.
6: return $S' = S \cup U$.
```

4.2. Correctness and Guarantees

Algorithm 1 satisfies the following guarantees. All omitted proofs can be found in Appendix B.

[6]We note that u, v, v' and w in the definitions below are mutually distinct. We omit writing this for simplicity.

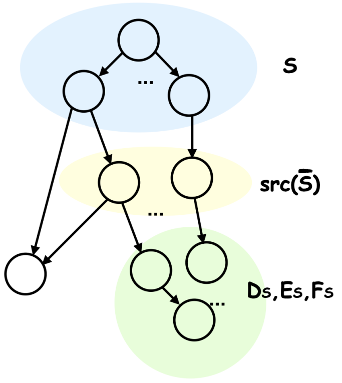

Figure 3. D_S, E_S, F_S satisfy that (1) they contain all downstream vertices of any vertex in them; (2) they do not intersect with $\text{src}(\bar{S})$.

Theorem 4.4. *Algorithm 1 outputs a prefix vertex set in $\mathcal{O}(n^4)$ number of CI tests. In addition, this prefix vertex set contains all the remaining source nodes, i.e., $\text{src}(\bar{S}) \subset S'$.*

For its proof, we will make use of the following properties of the type-I, II, and III sets (illustrated in Figure 3).

Lemma 4.5. *Let S be a prefix vertex set. If $w \in D_S$, then $\texttt{Des}[w] \subset D_S$. Furthermore, $D_S \cap \text{src}(\bar{S}) = \varnothing$. The same properties hold for E_S.*

Lemma 4.6. *Let S be a prefix vertex set. If $w \in F_S$, then $\texttt{Des}[w] \subset E_S \cup F_S$. Furthermore, $F_S \cap \text{src}(\bar{S}) = \varnothing$.*

Proof of Theorem 4.4. We first show that S' returned by Algorithm 1 satisfies $\text{src}(\bar{S}) \subset S'$. By Lemmas 4.5 and 4.6, we have $(D_S \cup E_S \cup F_S) \cap \text{src}(\bar{S}) = \varnothing$. As $\bar{S}' = D_S \cup E_S \cup F_S$, it must hold that $\text{src}(\bar{S}) \subset S'$.

Next we show that S' is a prefix vertex set. For this, we only need to show that $\forall w \in \bar{S}'$ and $y \in \texttt{Des}(w)$, it holds that $y \in \bar{S}'$. Since $\bar{S}' = D_S \cup E_S \cup F_S$, one of the following three scenarios must hold: (1) if $w \in D_S$, then $y \in D_S$ by Lemma 4.5; (2) if $w \in E_S$, then $y \in E_S$ by Lemma 4.5; or (3) if $w \in F_S$, then $y \in E_S \cup F_S$ by Lemma 4.6.

Therefore S' must be a prefix vertex set. We now bound the number of CI tests performed by Algorithm 1: computing D_S takes $\mathcal{O}(n^3)$ CI tests; computing E_S takes $\mathcal{O}(n^4)$ CI tests; computing F_S takes $\mathcal{O}(n^3)$ CI tests, which completes the proof. □

4.3. Relation to Covered Edges

In Theorem 4.4, we showed that $\text{src}(\bar{S}) \subset S'$. In fact, when there are no covered edges coming from $\text{src}(\bar{S})$, one can show that $S' = \text{src}(\bar{S}) \cup S$ via the following Lemma 4.7.

Lemma 4.7. *Let S be a prefix vertex set. For $w \in \bar{S} \setminus D_S$, if $w \notin \text{src}(\bar{S})$ and there is* no covered edge *from $\texttt{Anc}[w] \cap \text{src}(\bar{S})$ to $\texttt{Anc}[w]$, then $w \in E_S \cup F_S$.*

Corollary 4.8. *Let S be a prefix vertex set. If there is* no *covered edge in $\bar{S}$, then $S' = \mathrm{src}(\bar{S}) \cup S$.*

This result will be useful when deriving CCPG representations using Algorithm 1. To see this, consider the simple case where there is no covered edge in $\mathcal{G}$. Then running Algorithm 1 with $S = \varnothing$ we can learn $\mathrm{src}(V)$ by Corollary 4.8. Then running Algorithm 1 with $S = \mathrm{src}(V)$, we can learn the source vertices of $V \setminus \mathrm{src}(V)$. Applying this iteratively, we can obtain the ground-truth topological order of $\mathcal{G}$. As a consequence, one can easily learn $\mathcal{G}$,[7] which is the sole CCPG representation as there is no covered edge.

5. Learning Causally Consistent Partition Graph Representations

We now present our algorithm for learning causally consistent partition graph representations. All omitted proofs can be found in Appendix C.

Algorithm 2 contains three parts, indicated by colored boxes below. In the first part, we use Algorithm 1 iteratively to learn prefix vertex sets of the form $\varnothing \subsetneq \dots S \subsetneq S' \subsetneq \cdots \subsetneq V$. In the second part, we break each $S' \setminus S$ into smaller components that are pairwise independent given S. As we will see, these components satisfy the property of CCPG in Definition 3.1. In the third part, we build the acyclic graph between the CCPG components.

Algorithm 2 Learning a CCPG Representation

```
1: Input: CI queries from $\mathcal{G}$.
2: Output: A CCPG representation of $\mathcal{G}$.
3: Set $S = \varnothing$ and $\mathcal{S}$ as empty ordered list.
4: while $S \neq V$ do
5:     Run Algorithm 1 on $S$ to obtain $S'$.
6:     Add $S' \setminus S$ to the end of $\mathcal{S}$.
7:     Update $S = S'$.
8: end while
9: Initialize $l_1 = 1$.
10: for $S_i$ in $\mathcal{S} = [S_1, \dots, S_m]$ do
11:     Create empty graph $\mathcal{T}$ on vertices in $S_i$.
12:     Add $v - w$ to $\mathcal{T}$ iff $v \not\perp w \mid S_1 \cup \cdots \cup S_{i-1}$.
13:     Split $S_i$ into components $V_{l_i}, V_{l_i+1}, \dots, V_{l_{i+1}}$ based
        on connected components in $\mathcal{T}$.
14: end for
15: Create empty graph $\mathcal{D}$ on vertices in $[l_{m+1}]$.
16: for $V_i, V_j$ with $i < j$ do
17:     Add $i \to j$ to $\mathcal{D}$ iff $V_i \not\perp V_j \mid V_1 \cup \cdots \cup V_{j-1}$.
18: end for
19: return $V_1, \dots, V_{l_{m+1}}$ and $\mathcal{D}$
```

To show that the components we obtain in the second part

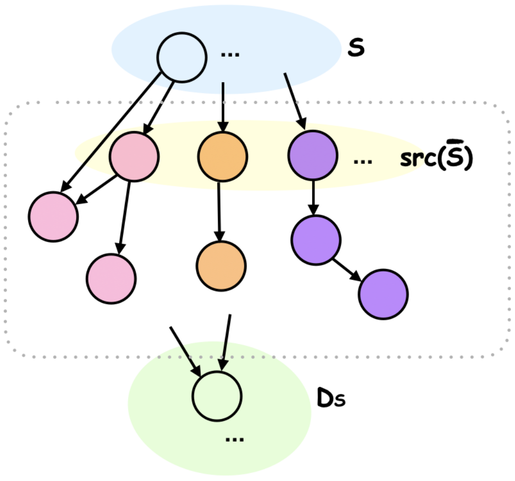

Figure 4. Illustration of $\bar{S} \setminus D_S$ (inside the dashed box). It can be split into connected subgraphs based on vertices in $\mathrm{src}(\bar{S})$ (indicated by the fill color of each vertex in $\bar{S} \setminus D_S$).

satisfy the property of CCPG, we will use the following result. Essentially, we can show that the subgraph $\mathcal{G}[S' \setminus S]$ can be split into connected subgraphs based on individual vertices in $\mathrm{src}(S' \setminus S)$ ($= \mathrm{src}(\bar{S})$ by Theorem 4.4). This is illustrated in Figure 4.

Lemma 5.1. *Let S be a prefix subset. For any $w \in \bar{S} \setminus D_S$, there is $|\mathtt{Anc}[w] \cap \mathrm{src}(\bar{S})| = 1$. Furthermore, for any other $w' \in \bar{S} \setminus D_S$, there is $w \not\perp w' \mid S$* if and only if *$\mathtt{Anc}[w] \cap \mathrm{src}(\bar{S}) = \mathtt{Anc}[w'] \cap \mathrm{src}(\bar{S})$.*

Based on this result, we can establish that Algorithm 2 outputs a CCPG representation by proving Theorem 3.3.

Proof of Theorem 3.3. We show that Algorithm 2 outputs a CCPG representation of $\mathcal{G}$. Since it runs in polynomial time and uses $\mathcal{O}(n^5)$ CI tests, this will prove Theorem 3.3.

Intra-component Property. Consider S_i. Denote $S_1 \cup \cdots \cup S_{i-1} = S$. We will show that S_i is split into $\mathtt{Des}[s] \cap S_i$ for each $s \in \mathrm{src}(\bar{S})$. Since $S_i \subset \bar{S}$, we have $S_i = \cup_{s \in \mathrm{src}(\bar{S})}(\mathtt{Des}[s] \cap S_i)$. By Theorem 4.4, we know that S is a prefix vertex set. Since $S_i \subset \bar{S} \setminus D_S$, by Lemma 5.1, $\mathtt{Des}[s] \cap S_i$ for different $s \in \mathrm{src}(\bar{S})$ are disjoint. Furthermore, by Theorem 4.4, these disjoint sets are non-empty because $s \in \mathtt{Des}[s] \cap S_i$. Therefore $\mathrm{src}(\mathtt{Des}[s] \cap S_i) = \{s\}$ and we have $|\mathrm{src}(\mathtt{Des}[s] \cap S_i)| = 1$. Thus by Lemma 5.1, each of $V_{l_i}, V_{l_i+1}, \dots, V_{l_{i+1}}$ corresponds to $\mathtt{Des}[s] \cap S_i$ for some $s \in \mathrm{src}(\bar{S})$.

If $|\mathtt{Des}[s] \cap S_i| > 1$, then it contains a vertex that is not s. Suppose $w \in \mathtt{Des}(s) \cap S_i$. By Lemmas 4.7 and 5.1, there is a covered edge from s to some vertex $x \in \mathtt{Anc}[w]$. Since S and $S \cup S_i$ are prefix vertex sets, $s, w \in S_i$ implies $x \in S_i$. Thus $x \in \mathtt{Des}[s] \cap S_i$ and there is a covered edge $s \to x$ in

[7] For $i < j$ in the topological order, the corresponding edge $v_i \to v_j \in \mathcal{G}$ iff $v_i \not\perp v_j \mid v_1, \dots, v_{j-1}$.

$\texttt{Des}[s] \cap S_i$. Thus, we have proven that each component in $V_1, \ldots, V_{l_{m+1}}$ satisfies the first property of CCPG.

Inter-component Property. For each V_i, denote the subset that it came from as S_{h_i}, i.e., $V_i \subset S_{h_i}$. By the construction of $\mathcal{D}$ in Algorithm 2, no edge $i \to j$ in $\mathcal{D}$ means either $i > j$ or $V_i \perp V_j \mid V_1 \cup \cdots \cup V_{j-1}$. If $i > j$ and $h_i \neq h_j$, then by the fact that $S_1 \cup \cdots \cup S_{h_i}$ is a prefix vertex set, there is no edge from V_i to V_j in $\mathcal{G}$. If $h_i = h_j = h$, then we know that there exists $s_i \neq s_j$ such that $V_i = \texttt{Des}[s_i] \cap S_h$ and $V_j = \texttt{Des}[s_j] \cap S_h$. If there is an edge from V_i to V_j in $\mathcal{G}$, denoted as $v \to w$, then $w \in \texttt{Des}[v] \subset \texttt{Des}[s_i]$ which means $V_i \cap V_j \neq \varnothing$, a contradiction. Therefore there is no edge from V_i to V_j in $\mathcal{G}$. If $V_i \perp V_j \mid V_1 \cup \cdots \cup V_{j-1}$, then clearly there is no edge from V_i to V_j in $\mathcal{G}$. Thus when there is no $i \to j$ in $\mathcal{D}$, there is no edge from V_i to V_j in $\mathcal{G}$.

If there is an edge $i \to j$ in $\mathcal{D}$, then we have $i < j$ and $V_i \not\perp V_j \mid V_1 \cup \cdots \cup V_{j-1}$. Note that since $S_1 \cup \cdots \cup S_{h_j}$ is prefixed, it holds that $\texttt{Pa}(V_j) \subset S_1 \cup \cdots \cup S_{h_j}$ and $\texttt{Des}[V_j] \cap (S_1 \cup \ldots S_{h_j - 1}) = \varnothing$. As shown above, there is no edge between any other $V_{j'} \subset S_{h_j}$ and V_j. Thus we have $\texttt{Pa}(V_j) \subset V_1 \cup \cdots \cup V_j$ and $\texttt{Des}[V_j] \cap (V_1 \cup \cdots \cup V_{j-1}) = \varnothing$. Similarly $\texttt{Pa}(V_i) \subset V_1 \cup \cdots \cup V_i$. Thus by the local Markov property and Bayes rule,[8] $V_i \not\perp V_j \mid V_1 \cup \cdots \cup V_{j-1}$ only if there is a direct edge $u \to v$ from $u \in V_i$ to $v \in V_j$. We now show that there is also an edge $u \to s$ where $\{s\} = \mathrm{src}(V_j)$. Assume on the contrary that there is no edge $u \to s$. Since s is the source node of V_j and thus a source node of S_{h_j}, we have from $\texttt{Pa}(V_j) \subset S_1 \cup \cdots \cup S_{h_j}$ that $\texttt{Pa}(s) \subset S_1 \cup \cdots \cup S_{h_j - 1}$. In addition, $\texttt{Des}[s] \cap (S_1 \cup \cdots \cup S_{h_j - 1}) = \varnothing$. Thus by the local Markov property, $u \perp s \mid S_1 \cup \cdots \cup S_{h_j - 1}$. However as $u \to v$ and $v \in \texttt{Des}[s]$, we have $u \not\perp s \mid S_1 \cup \cdots \cup S_{h_j - 1} \cup \{v\}$. As a consequence, $v \in D_{S_1 \cup \cdots \cup S_{h_j - 1}}$. By Algorithm 1, it is impossible that $v \in S_{h_j}$, a contradiction. Therefore we must have $u \in \texttt{Pa}(s)$. This proves that $\mathcal{D}$ satisfies the second property of CCPG. □

6. Interventions

In Sections 4 and 5, we showed how to learn a CCPG representation using observational data. Here we generalize these methods to interventions. This results in a more refined $\mathcal{I}$-CCPG representation of $\mathcal{G}$. All omitted proofs can be found in Appendix D.

6.1. Learning Refined Prefix Vertex Sets

In Section 4.1, we obtained a prefix vertex set by excluding three types of sets D_S, E_S and F_S. With interventions, we can exclude an additional set. To define it, we first characterize what can be learned using interventions.

[8] See Lemma A.1 in Appendix A.

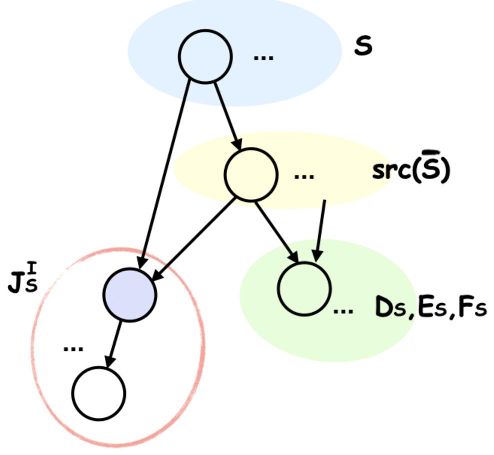

Figure 5. Illustration of J_S^I, where I is indicated by the purple circle. J_S^I satisfies similar properties as D_S, E_S and F_S.

Lemma 6.1. *Let S be a prefixed vertex set. Given an intervention on vertices $I \subseteq V$. For any $u \notin I$, we have $u \not\perp_I v$ for some $v \in I \setminus S$ if and only if $u \in \texttt{Des}(I \setminus S) \setminus (I \setminus S)$.*

Note that when the number of CI tests is not restricted, one can learn all edges cut by I as well as their orientations. Lemma 6.1 shows that it is possible to learn the joint set of descendants using $\mathcal{O}(n^2)$ CI tests.

With this set and a few more CI tests, it is possible to learn additional directional information.

Lemma 6.2. *Let S be a prefix subset. Given an intervention I and $v \in I \setminus S$, denote $H_S^I(v) = \{u \notin S \cup \texttt{Des}[I \setminus S] : u \not\perp v \mid V \setminus \texttt{Des}[I \setminus S]\}$.[9] Then $Pa(v) \setminus (S \cup \texttt{Des}[I \setminus S]) \subseteq H_S^I(v) \subseteq \texttt{Anc}(v) \setminus (S \cup \texttt{Des}[I \setminus S])$.*

Such sets can be useful when deciding if a target of $I \setminus S$ should be excluded when learning a prefix vertex subset $S' \supset S$. Formally, we define the type-IV set as follows.

Definition 6.3 (Type-IV Set J_S^I). Let S be a prefix vertex set. For an intervention I, let $J_S^I = \texttt{Des}(I \setminus S) \cup \{v \in I \setminus S : H_S^I(v) \cap \bar{S} \neq \varnothing\}$.

Analogous to Lemma 4.5, we can show that J_S^I satisfies similar properties (illustrated in Figure 5).

Lemma 6.4. *Let S be a prefix vertex set. Then $\forall w \in J_S^I$, we have $\texttt{Des}(w) \subseteq J_S^I$. Furthermore, $J_S^I \cap \mathrm{src}(\bar{S}) = \varnothing$.*

Therefore we can exclude J_S^I as well, which results in the following modification of Algorithm 1 and for which we can show similar guarantees using Lemma 6.4.

[9] Note that $\texttt{Des}[I \setminus S]$ can be obtained via Lemma 6.1 and $(\texttt{Des}(I \setminus S) \setminus (I \setminus S)) \cup (I \setminus S)$.

Algorithm 3 Learning a Prefix Vertex Set (w. Interventions)

```
1: Input: A prefix vertex set $S \subsetneq V$. CI queries from $\mathcal{G}$ and $\mathcal{G}^I$ for each intervention $I \in \mathcal{I}$.
2: Output: A prefix vertex set $S'$ such that $S' \supsetneq S$.
3: for $I \in \mathcal{I}$ do
4:     Compute type-IV set $J_S^I$.
5: end for
6: Compute type-I set $D_S$.
7: Compute type-II and III sets $E_S, F_S$.
8: Let $U' = \bar{S} \setminus \big(\cup_{I\in\mathcal{I}} J_S^I \cup D_S \cup E_S \cup F_S\big)$.
9: return $S' = S \cup U'$.
```

Theorem 6.5. *Algorithm 3 outputs a prefix vertex set in $\mathcal{O}(n^4)+|\mathcal{I}|\mathcal{O}(n^2)$ CI tests. In addition, this prefix vertex set contains all the remaining source nodes, i.e.,* $\mathrm{src}(\bar{S}) \subset S'$.

Proof. Similar to the proof of Theorem 3.3, we can use the additional Lemma 6.4 to show that: (1) S' is a prefix vertex set, (2) it contains $\mathrm{src}(\bar{S})$. We now bound the number of CI tests performed by Algorithm 3: note that it bears the additional need to compute J_S^I compared to Algorithm 1. By Lemmas 6.1 and 6.2, computing J_S^I for each $I \in \mathcal{I}$ takes $\mathcal{O}(n^2)$. Thus the total complexity is $\mathcal{O}(n^4)+|\mathcal{I}|\mathcal{O}(n^2)$. □

In addition, we can show the following property for covered edges coming from $\mathrm{src}(\bar{S})$.

Lemma 6.6. *Let S be a prefix vertex set, $v \in \mathrm{src}(\bar{S})$ and $v \to w$ be a covered edge. Given any intervention I, if $|I \cap \{v, w\}| = 1$, then $w \in J_S^I$.*

Proof. If $I \cap \{v, w\} = \{v\}$, then $w \in \mathtt{Des}(I \setminus S)$. If $I \cap \{v, w\} = \{w\}$, then $w \in I \setminus S$. We will show that $H_S^I(w) \cap \bar{S} \neq \varnothing$. This in turn proves that $w \in J_S^I$. Note that since $v \in \mathrm{src}(\bar{S})$, we have $v \notin S \cup \mathtt{Des}[I \setminus S]$. Furthermore, since $v \to w$ is an edge, we have $v \in \mathtt{Pa}(w)$. Thus by Lemma 6.2, we have $v \in \mathtt{Pa}(w) \setminus (S \cup \mathtt{Des}[I \setminus S]) \subseteq H_S^I(w)$. □

6.2. Learning $\mathcal{I}$-CCPG Representations

We obtain the final algorithm by plugging Algorithm 3 into Algorithm 2, i.e., replacing Algorithm 1 in line 5 by Algorithm 3.

Proof of Theorem 3.4. By Theorem 6.5, for any i, set $S = S_1 \cup \cdots \cup S_{i-1}$ is a prefix vertex set. Therefore using the proof in Theorem 3.3 and the fact that $\mathrm{src}(\bar{S}) \subset S \cup S_i$ (Theorem 6.5), we immediately have $|\mathrm{src}(V_i)| = 1$ and $\mathcal{D}$ satisfies the second property of $\mathcal{I}$-CCPG in Definition 3.1.

We now show that when $|V_i| > 1$, subgraph $\mathcal{G}[V_i]$ has at least one unintervened covered edge. Suppose on the contrary that $\mathcal{G}[V_i]$ has no unintervened covered edge. Denote $\{s\} = \mathrm{src}(V_i)$. Let $w \in V_i$ such that $w \neq s$. Similar to Theorem 3.3, we can show that $w \in \mathtt{Des}(s)$. By Lemma 4.7, there is a covered edge from s to some vertex $x \in \mathtt{Anc}[w]$ and $x \in V_i$. Since $\mathcal{G}[V_i]$ has no unintervened covered edge, $s \to x$ is intervened by some $I \in \mathcal{I}$. Then by Lemma 6.6, we have $x \in J_S^I$. This contradicts $x \in V_i$. □

7. Experiments

In this section, we test our proposed method and compare it to existing causal discovery methods on synthetic data and a toy real-world example. Source code for these results can be found at https://github.com/uhlerlab/CCPG.

7.1. Synthetic Data

In these experiments, we consider the observational setting and generate samples from linear causal models with additive Gaussian noise, governed by identifiable causal graphs, in particular, in-star-shaped DAGs with varying number of nodes. In such settings, with enough samples, our algorithm is guaranteed to return the ground-truth DAG (Corollary 3.5).

We compare our Algorithm 2, termed `CCPG`, to other constraint-based and hybrid methods that rely on conditional independence tests. The constraint-based methods we include are: `PC` (Spirtes et al., 2000), `FCI` (Spirtes et al., 1999), and `RFCI` (Colombo et al., 2012), where we use the order-independent variants from Colombo et al. (2014) (i.e., "`stable`" versions). The hybrid methods we included are: `GSP` with depth $= 4$ and depth $= \infty$ (Solus et al., 2021). Implementation details can be found in Appendix E.

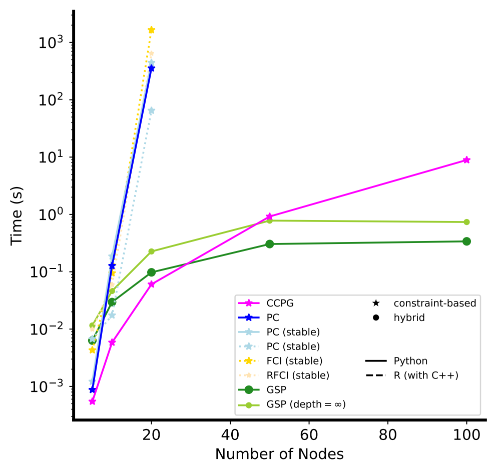

Figure 6. **Runtime comparison across graphs of different sizes.** `CCPG` is as fast as hybrid methods (`GSP`) and significantly more efficient than constraint-based methods (`PC`, `FCI`, and `RFCI`). The programming language behind each implementation is indicated by dashed or solid lines (see Appendix E for details).

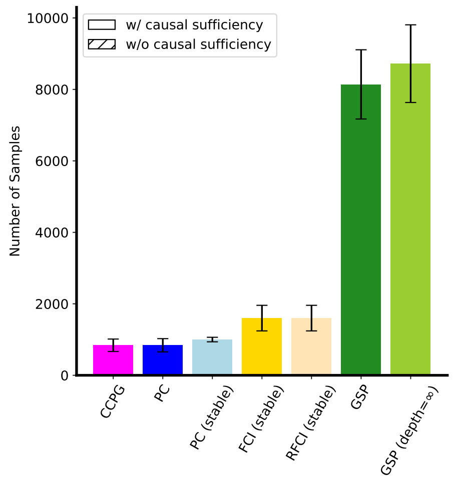

Figure 7. **Sample complexity comparison.** `CCPG` requires the least number of samples to recover the true graph. Methods that do not assume causal sufficiency are indicated by shading (see Appendix E for details).

Runtime Analysis. To test the computational efficiency of our method, we generated 100k samples across graphs of different sizes and reported the runtime averaged across five runs. Figure 6 shows that `CCPG` is as fast as hybrid methods (`GSP`) and significantly more efficient than constraint-based methods (`PC`, `FCI`, and `RFCI`), which can only scale to 20 nodes with reasonable runtime.

Sample Complexity Analysis. To test the sample efficiency of `CCPG`, we consider a 10-node in-star-shaped DAG. For each method, we increase the number of samples until it returns the ground-truth DAG and repeat this procedure for five runs. Figure 7 shows that `CCPG` requires the least number of samples to recover the true graph.

In general, constraint-based methods (`PC`, `FCI`, and `RFCI`) have better sample efficiency than hybrid methods, while being significantly more runtime inefficient. In comparison, `CCPG`, with its polynomial-number of CI test guarantee, enjoys low sample complexity and low runtime complexity.

7.2. A Real-world Example

To illustrate the utility of the coarser representation learned by `CCPG` in real-world settings, we include a simple 6-variable Airfoil example (Asuncion & Newman, 2007; Lam et al., 2022); see Lam et al. (2022) for a detailed description of this example. Although there is no known ground-truth DAG in this setting, a few causal relations are known: (1) velocity, chord, and attack should be source nodes; (2) pressure is downstream of all other nodes.

The coarser representation learned by `CCPG` is shown in Figure 8. Compared to `PC`, which returns the partially directed

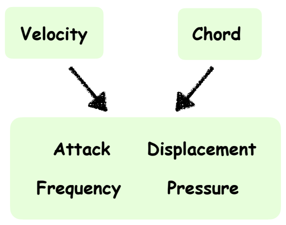

Figure 8. Learned coarser representation by `CCPG` in the Airfoil example.

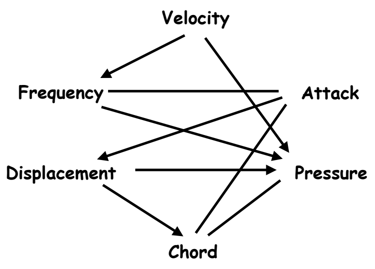

Figure 9. Learned partially directed graph by `PC` in the Airfoil example.

graph in Figure 9, `CCPG` seems to be more consistent with the known causal relations while it contains less information due to a coarser representation.

8. Discussion

In our work, we studied causal structure learning under the constraint of fewer CI tests. Since exact structure learning may demand an exponential number of CI tests, we defined a representation (CCPG) that captures partial but crucial information about the underlying causal graph. Moreover, we provided an efficient algorithm that recovers a CCPG representation in a polynomial number of CI tests. This result enabled us to design efficient algorithms for the full recovery of causal graphs in two specific settings, utilizing only a polynomial number of CI tests.

We hope that our work will motivate further exploration of the causal discovery problem under the constraint of fewer CI tests, extending to various settings, including those involving latent variables. Furthermore, our research establishes a foundation for addressing the search problem[10] with reduced tests, suggesting the potential existence of search algorithms capable of recovering the causal graph with a polynomial number of independence tests while performing an approximately optimal number of interventions.

[10]The search problem involves finding the minimum set of interventions that orient the entire causal graph.

Acknowledgements

We thank the anonymous reviewers for helpful feedback. J.Z. was partially supported by an Apple AI/ML PhD Fellowship. K.S. was supported by a fellowship from the Eric and Wendy Schmidt Center at the Broad Institute. C.U. was partially supported by NCCIH/NIH (1DP2AT012345), ONR (N00014-22-1-2116), DOE-ASCR (DE-SC0023187), the MIT-IBM Watson AI Lab, and a Simons Investigator Award.

Impact Statement

This paper presents theoretical work whose goal is to advance the field of causal inference. There are many potential societal applications of our work, none of which we feel must be specifically highlighted here.

References

Alonso-Barba, J. I., Gámez, J. A., Puerta, J. M., et al. Scaling up the greedy equivalence search algorithm by constraining the search space of equivalence classes. *International journal of approximate reasoning*, 54(4):429–451, 2013.

Andersson, S. A., Madigan, D., and Perlman, M. D. A characterization of Markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.

Asuncion, A. and Newman, D. Uci machine learning repository, 2007.

Bouckaert, R. R. Properties of bayesian belief network learning algorithms. In *Uncertainty Proceedings 1994*, pp. 102–109. Elsevier, 1994.

Brenner, E. and Sontag, D. Sparsityboost: A new scoring function for learning bayesian network structure. *arXiv preprint arXiv:1309.6820*, 2013.

Chickering, D. M. A Transformational Characterization of Equivalent Bayesian Network Structures. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pp. 87–98, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

Chickering, D. M. Optimal Structure Identification with Greedy Search. *Journal of machine learning research*, 3 (Nov):507–554, 2002.

Chickering, M., Heckerman, D., and Meek, C. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004.

Cho, H., Berger, B., and Peng, J. Reconstructing Causal Biological Networks through Active Learning. *PLoS ONE*, 11(3):e0150611, 2016.

Choo, D. and Shiragur, K. Subset verification and search algorithms for causal dags. *arXiv preprint arXiv:2301.03180*, 2023.

Choo, D., Shiragur, K., and Bhattacharyya, A. Verification and search algorithms for causal DAGs. *Advances in Neural Information Processing Systems*, 35, 2022.

Claassen, T., Mooij, J., and Heskes, T. Learning sparse causal models is not np-hard. *arXiv preprint arXiv:1309.6824*, 2013.

Colombo, D., Maathuis, M. H., Kalisch, M., and Richardson, T. S. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pp. 294–321, 2012.

Colombo, D., Maathuis, M. H., et al. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.*, 15(1):3741–3782, 2014.

de Campos, L. M., Cano, A., Castellano, J. G., and Moral, S. Combining gene expression data and prior knowledge for inferring gene regulatory networks via Bayesian networks using structural restrictions. *Statistical Applications in Genetics and Molecular Biology*, 18(3), 2019.

Eberhardt, F. Causal Discovery as a Game. In *Causality: Objectives and Assessment*, pp. 87–96. PMLR, 2010.

Eberhardt, F. and Scheines, R. Interventions and Causal Inference. *Philosophy of science*, 74(5):981–995, 2007.

Eberhardt, F., Glymour, C., and Scheines, R. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among N variables. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pp. 178–184, 2005.

Eberhardt, F., Glymour, C., and Scheines, R. N-1 Experiments Suffice to Determine the Causal Relations Among N Variables. In *Innovations in machine learning*, pp. 97–112. Springer, 2006.

Eberhardt, F., Glymour, C., and Scheines, R. On the Number of Experiments Sufficient and in the Worst Case Necessary to Identify All Causal Relations Among N Variables. *arXiv preprint arXiv:1207.1389*, 2012.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

Geiger, D. and Heckerman, D. Parameter priors for directed acyclic graphical models and the characterization of several probability distributions. *The Annals of Statistics*, 30 (5):1412–1440, 2002.

Geiger, D. and Pearl, J. On the logic of causal models. In *Machine Intelligence and Pattern Recognition*, volume 9, pp. 3–14. Elsevier, 1990.

Glymour, C., Zhang, K., and Spirtes, P. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Greenewald, K., Katz, D., Shanmugam, K., Magliacane, S., Kocaoglu, M., Boix-Adserà, E., and Bresler, G. Sample Efficient Active Learning of Causal Trees. *Advances in Neural Information Processing Systems*, 32, 2019.

Hauser, A. and Bühlmann, P. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1):2409–2464, 2012.

Hoover, K. D. The logic of causal inference: Econometrics and the Conditional Analysis of Causation. *Economics & Philosophy*, 6(2):207–234, 1990.

Hu, H., Li, Z., and Vetta, A. Randomized Experimental Design for Causal Graph Discovery. *Advances in neural information processing systems*, 27, 2014.

Kalisch, M. and Bühlman, P. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8(3), 2007.

Kalisch, M., Hauser, A., Maechler, M., Colombo, D., Entner, D., Hoyer, P., Hyttinen, A., Peters, J., Andri, N., Perkovic, E., et al. Package 'pcalg'. 2024.

King, R. D., Whelan, K. E., Jones, F. M., Reiser, P. G. K., Bryant, C. H., Muggleton, S. H., Kell, D. B., and Oliver, S. G. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971): 247–252, 2004.

Lam, W.-Y., Andrews, B., and Ramsey, J. Greedy relaxations of the sparsest permutation algorithm. In *Uncertainty in Artificial Intelligence*, pp. 1052–1062. PMLR, 2022.

Lauritzen, S. L. *Graphical models*, volume 17. Clarendon Press, 1996.

Magliacane, S., Claassen, T., and Mooij, J. M. Ancestral causal inference. *Advances in Neural Information Processing Systems*, 29, 2016.

Meek, C. Causal Inference and Causal Explanation with Background Knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pp. 403–410, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

Nandy, P., Hauser, A., and Maathuis, M. H. High-dimensional consistency in score-based and hybrid structure learning. *The Annals of Statistics*, 46(6A):3151–3183, 2018.

Pearl, J. Causality: models, reasoning, and inference. *Econometric Theory*, 19(4):675–685, 2003.

Pearl, J. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

Pingault, J.-B., O'reilly, P. F., Schoeler, T., Ploubidis, G. B., Rijsdijk, F., and Dudbridge, F. Using genetic data to strengthen causal inference in observational research. *Nature Reviews Genetics*, 19(9):566–580, 2018.

Reichenbach, H. *The direction of time*, volume 65. Univ of California Press, 1956.

Robins, J. M., Hernan, M. A., and Brumback, B. Marginal structural models and causal inference in epidemiology. *Epidemiology*, pp. 550–560, 2000.

Rotmensch, M., Halpern, Y., Tlimat, A., Horng, S., and Sontag, D. Learning a Health Knowledge Graph from Electronic Medical Records. *Scientific reports*, 7(1):1–11, 2017.

Schulte, O., Frigo, G., Greiner, R., and Khosravi, H. The imap hybrid method for learning gaussian bayes nets. In *Advances in Artificial Intelligence: 23rd Canadian Conference on Artificial Intelligence, Canadian AI 2010, Ottawa, Canada, May 31–June 2, 2010. Proceedings 23*, pp. 123–134. Springer, 2010.

Shanmugam, K., Kocaoglu, M., Dimakis, A. G., and Vishwanath, S. Learning Causal Graphs with Small Interventions. *Advances in Neural Information Processing Systems*, 28, 2015.

Shiragur, K., Zhang, J., and Uhler, C. Meek separators and their applications in targeted causal discovery. *Advances in Neural Information Processing Systems*, 36, 2024.

Solus, L., Wang, Y., and Uhler, C. Consistency guarantees for greedy permutation-based causal inference algorithms. *Biometrika*, 108(4):795–814, 2021.

Spirtes, P., Glymour, C., and Scheines, R. Causality from probability. 1989.

Spirtes, P., Meek, C., and Richardson, T. An algorithm for causal inference in the presence of latent variables and selection bias (vol. 1), 1999.

Spirtes, P., Glymour, C. N., Scheines, R., and Heckerman, D. *Causation, Prediction, and Search*. MIT press, 2000.

Spirtes, P. L., Meek, C., and Richardson, T. S. Causal inference in the presence of latent variables and selection bias. *arXiv preprint arXiv:1302.4983*, 2013.

Squires, C. Causaldag: Creation, manipulation, and learning of causal models. 2018. *URL https://github.com/uhlerlab/causaldag*.

Squires, C. and Uhler, C. Causal structure learning: a combinatorial perspective. *Foundations of Computational Mathematics*, pp. 1–35, 2022.

Squires, C., Magliacane, S., Greenewald, K., Katz, D., Kocaoglu, M., and Shanmugam, K. Active Structure Learning of Causal DAGs via Directed Clique Trees. *Advances in Neural Information Processing Systems*, 33:21500–21511, 2020.

Sverchkov, Y. and Craven, M. A review of active learning approaches to experimental design for uncovering biological networks. *PLoS computational biology*, 13(6): e1005466, 2017.

Tian, T. Bayesian Computation Methods for Inferring Regulatory Network Models Using Biomedical Data. *Translational Biomedical Informatics: A Precision Medicine Perspective*, pp. 289–307, 2016.

Verma, T. and Pearl, J. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 255–270, 1990.

Woodward, J. *Making Things Happen: A theory of Causal Explanation*. Oxford university press, 2005.

Xie, X. and Geng, Z. A recursive method for structural learning of directed acyclic graphs. *The Journal of Machine Learning Research*, 9:459–483, 2008.

Zhang, J., Shiragur, K., and Uhler, C. Membership testing in markov equivalence classes via independence queries. In *International Conference on Artificial Intelligence and Statistics*, pp. 3925–3933. PMLR, 2024.

A. Useful Lemmas

A.1. Proof of Lemma 3.2

Lemma 3.2 (Properties of CCPG). *For any intervention set $\mathcal{I}$ (including $\varnothing$), the following arguments hold:*

- *$\mathcal{D} = \mathcal{G}$ (i.e., partitioning V into individual vertices) is a valid $\mathcal{I}$-CCPG of $\mathcal{G}$.*
- *If the verification number of $\mathcal{G}$ is zero, i.e., $\nu(\mathcal{G}) = 0$, then $\mathcal{D} = \mathcal{G}$ is the unique valid $\mathcal{I}$-CCPG of $\mathcal{G}$.*
- *If $\mathcal{I}$ is a verifying intervention set of $\mathcal{G}$, then $\mathcal{D} = \mathcal{G}$ is the unique valid $\mathcal{I}$-CCPG of $\mathcal{G}$.*

Proof. Note that if $|V_i| = 1$, then CCPG is actually $\mathcal{G}$. If $|V_i| > 1$, then it means that there is a covered edge in this subset that is not intervened. By Proposition 2.1, this is impossible when $\mathcal{I}$ is a verifying intervention set. □

A.2. Proof of Lemma 3.6

Lemma 3.6 (Proxy V-Structure). *Let $S \subseteq V$ and $u, v, z \in V \backslash S$. If $u \perp v | S$ and $u \not\perp v | S \cup \{z\}$, then $u, v \notin \texttt{Des}[z]$.*[11]

Proof. Let P be an active path (that carries dependency) between u and v when conditioned on $S \cup \{z\}$. Since $u \perp v|S$ and $u \not\perp v|S \cup \{z\}$, we get that there exist a set of vertices $w_1 \dots w_k$ that are colliders on P and satisfy: $\texttt{Des}[w_i] \cap S = \varnothing$ and $z \in \texttt{Des}[w_i]$.

Consider the path P and note that it takes the form $P = u - \cdots \to w_1 \leftarrow \cdots \to w_i \leftarrow \cdots \to w_k \leftarrow \cdots - v$. Define $P_1 = u - \cdots \to w_1$ and $P_2 = w_k \leftarrow \cdots - v$. Since all the colliders on paths P_1, P_2 are in or have their descendants in S and all non-colliders do not belong to S, we have that P_1, P_2 are active given S. We prove our lemma using the proof by contradiction strategy. For contradiction, let us assume that one of the vertices in $\{u, v\}$ belong to the set $\texttt{Des}[z]$ and without loss of generality let that vertex be u. Then since $z \in \texttt{Des}[w_i]$ for all i, there must be $u \in \texttt{Des}[w_i]$ for all i.

Since $u \in \texttt{Des}[w_k]$, let Q be the directed path in the graph that connects w_k to u. Note that all the vertices in path Q are all descendants of w_k and they do not belong to the set S (because $\texttt{Des}[w_k] \cap S = \varnothing$). Now consider the path (P_2, Q) that connects vertices v and u and note that the vertex w is a non-collider on the new path (P_2, Q). Since P_2 and Q are active paths given S and since $w \notin S$, we immediately get that the path (P_2, Q) is an active path given S, which further implies that $u \not\perp v|S$; a contradiction and we conclude the proof. □

A.3. Proof of Lemma 3.8

Lemma 3.8 (Proxy Meek Rule 1). *Let S be a prefix subset. If u, v and w are such that $u \in S$, $v, w \notin S$ and $u \not\perp v|S$ and $u \perp w|S \cup \{v\}$ then $v \notin \texttt{Des}[w]$.*

Proof. Suppose on the contrary that there is u, v, w such that $u \in S$, $v, w \notin S$ and $u \not\perp v|S$ and $u \perp w|S \cup \{v\}$ and $v \in \texttt{Des}[w]$.

Since $u \not\perp v \mid S$, let P be the active path connecting u and v given S. As $u \in S$ and $v \notin S$, there is an edge $u' - v'$ on P such that $u' \in S$ but $v' \notin S$.

Now denote the vertex on P that is immediate next to v as x. If $x \leftarrow v$, then there must be a collider on P between u' and v as S is a prefix subset where $u' \in S$ and $v \notin S$. Let y be the collider on P between u' and v that is closest to u', then $u' \to v' \cdots \to y$. Since P is active given S, the collider $y \in \texttt{Anc}[S]$. However $v' \in \texttt{Anc}[y] \subseteq \texttt{Anc}[S]$ and $v' \notin S$, a contradiction to S being prefix. Thus there must be $x \to v$ on P.

Since we assumed on the contrary that $v \in \texttt{Des}[w]$, we can consider the path Q joined by P and the directed path from w to v. Compared to P, the path P has one additional collider v, and has a few additional colliders that lie between v and w which are not in S (as they are all descendants of w and $w \notin S$). Therefore Q is active given $S \cup \{v\}$. This means $u \not\perp w \mid S \cup \{v\}$, a contradiction. □

[11]Similar argument is also proven in Lemma 1 of Magliacane et al. (2016).

A.4. Additional Lemma

In addition, we will make use of the following lemma.

Lemma A.1. *For disjoint sets $A, B, C \subset V$, if $\texttt{Pa}(A) \subset C \cup A$, $\texttt{Pa}(B) \subset C \cup B$ and $\texttt{Des}(B) \cap C = \varnothing$, then $A \perp B \mid C$.*

Proof. Assume without loss of generality that the vertices in B have the following topological order $b_1, \dots, b_m$. Then for any $i \in [m]$, by the local Markov property (Spirtes et al., 1989), we have $A \perp b_i \mid C \cup \{b_1, \dots, b_{i-1}\}$. Therefore using Bayes rule, we have

$$\begin{aligned} P(B \mid C, A) &= P(b_1 \mid C, A) P(b_2 \mid C, A, b_1) \dots P(b_m \mid C, A, b_1, \dots, b_{m-1}) \\ &= P(b_1 \mid C, A) P(b_2 \mid C, b_1) \dots P(b_m \mid C, b_1, \dots, b_{m-1}) \\ &= P(B \mid C), \end{aligned}$$

and thus $A \perp B \mid C$, which completes the proof. □

B. Missing Proofs of Prefix Vertex Set

B.1. Proof of Lemma 4.5

We restate the lemma below.

Lemma 4.5. *Let S be a prefix vertex set. If $w \in D_S$, then $\texttt{Des}[w] \subset D_S$. Furthermore, $D_S \cap \mathrm{src}(\bar{S}) = \varnothing$. The same properties hold for E_S.*

B.1.1. Type-I Set D_S

Proof of Lemma 4.5 for D_S. We first show that if $w \in D_S$, then it must hold that $\texttt{Des}[w] \subset D_S$: since $w \in D_S$, there exists a vertex $u \in V$ and $v \in \bar{S}$ such that $u \perp v \mid S$ and $u \not\perp v \mid S \cup \{w\}$. We now show that for any $x \in \texttt{Des}[w]$, we have $u \not\perp v \mid S \cup \{x\}$.

Since $u \not\perp v \mid S \cup \{w\}$, there is a path P from u to v that is active given $S \cup \{w\}$. Therefore, all non-colliders on P are not in $S \cup \{w\}$ and all colliders on P are in $\texttt{Anc}[S \cup \{w\}]$. Since $x \in \texttt{Des}[w]$, all colliders on P are in $\texttt{Anc}[S \cup \{x\}]$. If all non-colliders are not in $S \cup \{x\}$, then P is an active path from u to v given $S \cup \{x\}$, and thus $u \not\perp v \mid S \cup \{x\}$. Otherwise there is a non-collider on P that is x.

Since $u \perp v \mid S$, the path P is inactive given S. From above we know that all non-colliders on P are not in S. Therefore there exists a collider on P that is not in $\texttt{Anc}[S]$. Suppose the leftmost and rightmost such colliders are k, k' (it is possible that $k = k'$), then k, k' must be in $\texttt{Anc}[S \cup \{w\}] \setminus \texttt{Anc}[S] \subseteq \texttt{Anc}[w] \subset \texttt{Anc}[x]$. Consider the path Q in the graph by cutting out the parts between k, x (and k', x) on P and replacing them with directed edges from k to x (and from k' to x). Compared to P, the additional non-colliders on Q are all on the directed path from k to x (or k' to x). They are not in S since $k, k' \notin \texttt{Anc}[S]$, and thus Q has no non-colliders in S.

Compared to P, there is no collider on P that is not in $\texttt{Anc}[S]$ and is still on Q by the fact that k, k' are leftmost and rightmost colliders on P that are not in $\texttt{Anc}[S]$. Therefore, x must be a collider on Q, or else Q is active given S and $u \not\perp v \mid S$. Therefore all non-colliders on Q are not in $S \cup \{x\}$. Every collider on Q is either x or a collider of P, which is in $\texttt{Anc}[S \cup \{x\}]$. Thus Q is active given $S \cup \{x\}$. Therefore $u \not\perp v \mid S \cup \{x\}$.

Next we show that $D_S \cap \mathrm{src}(\bar{S}) = \varnothing$: for contradiction assume that there exists a vertex $a \in \mathrm{src}(\bar{S})$ such that $a \notin \bar{S} \backslash D_S$, that is $a \in \mathrm{src}(\bar{S})$ and for some vertex $u \in V, v \in \bar{S}$, $v \perp u | S$ and $v \not\perp u | S \cup \{a\}$.

Since $v \perp u | S$ and $v \not\perp u | S \cup \{a\}$, there exists a path P between v and u which is inactive when conditioned on S but is active upon conditioning on $S \cup \{a\}$. Moreover, this path contains a vertex b that is a collider on P and satisfies: $a \in \texttt{Des}[b]$ and $\texttt{Des}[b] \cap S = \varnothing$. Since $\texttt{Des}[b] \cap S = \varnothing$, we have that $b \in \bar{S}$. Furthermore, since $b \in \bar{S}, a \in \mathrm{src}(\bar{S})$ and $a \in \texttt{Des}[b]$, this implies that $b = a$. Therefore, the path P takes the form: $P = v \dots \to a \leftarrow \dots u$. All the colliders on the path P either belong to or have descendant in the set $S \cup \{a\}$.

Now consider the path $v \dots \to a$ and note that it is active given S. Let k be the number of vertices between v and a on this path $v - v_1 \dots v_k \to a$. It is immediate that $v_k \in S$ since $a \in \mathrm{src}(\bar{S})$. However, since $v_k \in S$, and since we condition on the set S, this should be a collider for the path Q to be active, which is not possible. Thus we get a contradiction, which completes the proof. $\square$

B.1.2. Type-II Set E_S

Proof of Lemma 4.5 for E_S. We first show that if $w \in E_S$, it must hold that $y \in E_S$ for any $y \in \texttt{Des}(w)$: since $w \in E_S$, there is $u \in S$ and $v, v' \in \bar{S} \setminus D_S$, such that $u \perp v' \mid S \cup \{v\}$ and $u \not\perp v' \mid S \cup \{v, w\}$. We will show $u \not\perp v' \mid S \cup \{v, y\}$.

Since $u \perp v' \mid S \cup \{v\}$ and $u \not\perp v' \mid S \cup \{v, w\}$, by Lemma 3.6 (note that the set "S" in the exposition of Lemma 3.6 can be an arbitrary subset), we know that $v' \notin \texttt{Des}[w]$. Assume on the contrary that $u \perp v' \mid S \cup \{v, y\}$. Since $u \perp v' \mid S \cup \{v\}$ and $u \not\perp v' \mid S \cup \{v, w\}$, there is a path P between u, v' that is active given $S \cup \{v, w\}$ but inactive given $S \cup \{v\}$ or $S \cup \{v, y\}$. This means that (1) y is a non-collider on P, (2) all colliders on P are in $\texttt{Anc}[S \cup \{v, w\}]$, (3) P has a collider that is in $\texttt{Anc}[w] \setminus \texttt{Anc}[S \cup \{v\}]$.

Note that $v' \notin \texttt{Des}[y]$ since $y \in \texttt{Des}[w]$ but $v' \notin \texttt{Des}[w]$. In addition, we also have $u \notin \texttt{Des}[w]$ since u is in the prefix vertex set S but $w \in \bar{S}$. Therefore y being a non-collider on P means there is a collider x on P such that $y \in \texttt{Anc}(x)$. Note that this collider has to be in $\texttt{Anc}[S \cup \{v, w\}]$. However, since $y \notin \texttt{Anc}[S \cup \{w\}]$ and $y \in \texttt{Anc}(x)$, it must hold that $x \in \texttt{Anc}[v]$, which means $y \in \texttt{Anc}(v)$. Since $y \in \texttt{Des}(w)$, this means $v \in \texttt{Des}(w)$, which makes $\texttt{Anc}[w] \setminus \texttt{Anc}[S \cup \{v\}] = \varnothing$. This violates (3) above, a contradiction.

Next we show that $E_S \cap \mathrm{src}(\bar{S}) = \varnothing$: if there exists $w \in \bar{S} \setminus D_S$ such that $w \in E_S \cap \mathrm{src}(\bar{S})$. Then $u \perp v' \mid S \cup \{v\}$ and $u \not\perp v' \mid S \cup \{v, w\}$ for some $u \in S$ and $v, v' \in \bar{S} \setminus D_S$. Thus there is a path P connecting u and v' such that P is active given $S \cup \{v, w\}$ but inactive given $S \cup \{v\}$.

Therefore all non-colliders on P are not in $S \cup \{v, w\}$, and there is a collider x on P such that $x \in \texttt{Anc}[w]$. Note that since $w \in \mathrm{src}(\bar{S})$, it must hold that $x \in S$. Since x is a collider, there is $y \neq u$ such that $y \to x$ on P. Since S is a prefix and $x \in S$, we have $y \in S$. However, y is a non-collider on P, which contradicts P active given $S \cup \{v, w\}$ and completes the proof. $\square$

B.2. Proof of Lemma 4.6

Lemma 4.6. *Let S be a prefix vertex set. If $w \in F_S$, then $\texttt{Des}[w] \subset E_S \cup F_S$. Furthermore, $F_S \cap \mathrm{src}(\bar{S}) = \varnothing$.*

Proof. We first show that if $w \in F_S$, then for any $y \in \texttt{Des}(w)$, we have $y \in E_S \cup F_S$. Since $w \in F_S$, we have $u \not\perp v \mid S, u \perp w \mid S \cup \{v\}$, and $v \not\perp w \mid S$ for some $u \in S$ and $v \in \bar{S} \setminus D_S$.

Since $v \not\perp w \mid S$, there is an active path between v, w given S. Consider extending this path by the directed path from w to y. Note that none of vertices on the directed path from w to y are in S, since S is prefixed and $w \notin S$. Therefore, this extended path is also active given S, which means $v \not\perp y \mid S$.

Thus, if $y \notin F_S$, then it must hold that $u \not\perp y \mid S \cup \{v\}$. This means there is an active path, denoted by P, between u and y given $S \cup \{v\}$. Consider extending this path by the directed path from w to y, denoted as Q (which exists in the graph). Compared to P, the additional non-colliders on Q are not in $S \cup \{v\}$: for S, this is because S is prefix, $w \notin S$, and all additional non-colliders are descendants of w; for v, this is because $v \notin \texttt{Des}(w)$ (by Lemma 3.8, $u \not\perp v \mid S$ and $u \perp w \mid S \cup \{v\}$). Thus Q is active given $S \cup \{v\}$ unless y is a collider on Q. Since $u \perp w \mid S \cup \{v\}$, the path Q must be inactive given $S \cup \{v\}$, which means y is a collider on Q. This means Q is active given $S \cup \{v, y\}$. Therefore, $u \not\perp w \mid S \cup \{v, y\}$. Together with $u \perp w \mid S \cup \{v\}$, we have $y \in E_S$.

Next we show that $F_S \cap \mathrm{src}(\bar{S}) = \varnothing$. Assume on the contrary that $w \in F_S \cap \mathrm{src}(\bar{S})$. Since $w \in F_S$, we have $u \not\perp v | S, v \not\perp w | S$ and $u \perp w | S \cup \{v\}$ for some $u \in S$ and $v \in \bar{S} \setminus D_S$. By Lemma 3.8, we have $v \notin \texttt{Des}[w]$. However, since $v \not\perp w \mid S$, there must be an active path P between v and w given S. This path cannot have any vertex in S; otherwise consider the first vertex that is in S; since $v \notin S$ and S is prefix, such vertex must be a non-collider which would make P inactive given S. Therefore P is fully in $\bar{S}$. Since $w \in \mathrm{src}(\bar{S})$, we must have $P : v - \cdots \leftarrow w$. However since $v \notin \texttt{Des}[w]$, there must be an edge $\to$ on P. This means that there must be a collider on P. Since this collider is not in S, it makes P inactive given S, a contradiction. $\square$

B.3. Remarks

The above proofs suffice as intermediate results to show Theorem 4.4, which we proved in Section 4.2.

Regarding Lemma 4.7 in Section 4.3, we will prove it in Appendix C after proving Lemma 5.1, since it depends on Lemma 5.1.

C. Missing Proofs of Causally Consistent Partition Graph Representations

C.1. Proof of Lemma 5.1

To prove Lemma 5.1, we will make use of the following lemma.

Lemma C.1. *Let $S \subseteq V$ be a prefix subset, then the following statements hold:*

- $u \perp v \mid S$ *for all* $u, v \in \mathrm{src}(\bar{S})$.
- $S \cup U$ *is a prefix subset for any* $U \subseteq \mathrm{src}(\bar{S})$.

Proof. We first prove condition one. For contradiction, we assume that $u \not\perp v|S$, which implies that there exists an active path between u and v when conditioned on S. Let $P = u - u_1 \dots u_k - v$ be the path and $u_1, \dots, u_k$ be the vertices along the path. Note that $k > 0$ as u and v are not connected; because an edge between u and v would mean that one of these vertices is not a source node in $\bar{S}$.

Consider u_1 and note that there are two possibilities $u \to u_1$ or $u \leftarrow u_1$. We start with the first case $u \to u_1$. Since S is a prefix subset and $u \in \bar{S}$, $u \to u_1$ implies that $u_1 \in \bar{S}$. Since no vertex in $\bar{S}$ is conditioned upon, we have that the vertex u_1 is not a collider on the path P and we have that the edge $u_1 - u_2$ is directed as $u_1 \to u_2$. Repeating the same argument for u_2 and all the other vertices in the path, we see that the path P takes the form $P = u \to u_1 \to \dots \to u_k \to v$. The previous argument implies that $v \in \mathtt{Des}[u]$ and therefore does not belong to $\mathrm{src}(\bar{S})$, which is a contradiction to our assumption that $u, v \in \mathrm{src}(\bar{S})$.

Now consider the other case where $u \leftarrow u_1$. Note that since $u \in \mathrm{src}(\bar{S})$, it is immediate that $u_1 \in S$. Furthermore, irrespective of the orientation between u_1 and vertex u_2, it holds that vertex u_1 is a non-collider on the path P. Moreover, since P is an active path, u_1 should not be conditioned upon. However, since we condition on S and as $u_1 \in S$, we have a contradiction.

In the above case analysis, we showed that there does not exist an active path between u and v when conditioned on S, which implies that $u \perp v|S$ and we conclude the proof for condition one.

In the remainder of the proof, we focus our attention on condition two. Consider any subset $U \subseteq \mathrm{src}(\bar{S})$. For contradiction assume that $S \cup U$ is not a prefix subset, which implies that there exists a vertex $v \in V \backslash (S \cup U)$ such that $v \in \mathtt{Anc}[u]$ for some vertex $u \in S \cup U$. Since S is a prefix subset, it is immediate that $u \notin S$. Therefore, the only case is that $u \in U$. Note that both u and v belong to the set $\bar{S}$ and $v \in \mathtt{Anc}(u)$; both these expressions combined contradict the fact that $u \in \mathrm{src}(\bar{S})$. Therefore, it should be the case that $S \cup U$ is a prefix subset and we conclude the proof. □

Now we prove Lemma 5.1 restated below.

Lemma 5.1. *Let S be a prefix subset. For any $w \in \bar{S} \backslash D_S$, there is $|\mathtt{Anc}[w] \cap \mathrm{src}(\bar{S})| = 1$. Furthermore, for any other $w' \in \bar{S} \setminus D_S$, there is $w \not\perp w' \mid S$* if and only if $\mathtt{Anc}[w] \cap \mathrm{src}(\bar{S}) = \mathtt{Anc}[w'] \cap \mathrm{src}(\bar{S})$.

Proof. We first show that for any $w \in \bar{S} \setminus D_S$, we have $|\mathtt{Anc}[w] \cap \mathrm{src}(\bar{S})| = 1$. Since $w \in \bar{S}$, it must hold that $|\mathtt{Anc}[w] \cap \mathrm{src}(\bar{S})| \geq 1$. Assume now that there is $w \in \bar{S}$ such that $|\mathtt{Anc}[w] \cap \mathrm{src}(\bar{S})| \geq 2$. Let $v_1, v_2 \in \mathtt{Anc}[w] \cap \mathrm{src}(\bar{S})$ such that $v_1 \neq v_2$. By Lemma C.1, we have $v_1 \perp v_2 \mid S$. Now consider the path P by stitching together the two directed paths, one from v_1 to w and another from v_2 to w. All non-colliders on P are not in S since S is a prefix subset. The only collider on P is w. Therefore P is active given $S \cup \{w\}$. We have $v_1 \not\perp v_2 \mid S \cup \{w\}$, which means $w \in D_{v_1,S} \subset D_S$, a contradiction to $w \notin D_S$.

Next we show that for any other $w' \in \bar{S} \setminus D_S$, we have $w \not\perp w' \mid S$ if and only if $\mathtt{Anc}[w] \cap \mathrm{src}(\bar{S}) = \mathtt{Anc}[w'] \cap \mathrm{src}(\bar{S})$. For the if direction, denote $s = \mathtt{Anc}[w] \cap \mathrm{src}(\bar{S}) = \mathtt{Anc}[w'] \cap \mathrm{src}(\bar{S})$ and consider the trek by joining the two directed paths from s to w and from s to w'. Since this path has no colliders and it is fully in $\bar{S}$ (since S is a prefix and $s \in \bar{S}$), it is active given S. Thus $w \not\perp w' \mid S$. For the only if direction, assume on the contrary that $w \not\perp w' \mid S$ but $\texttt{Anc}[w] \cap \mathrm{src}(\bar{S}) \neq \texttt{Anc}[w'] \cap \mathrm{src}(\bar{S})$. Let P be the active path between w, w' given S. Then all non-colliders on P are in $\bar{S}$ and all colliders on P are in $\texttt{Anc}[S] = S$. This means that P has no colliders; otherwise this collider is a child of the vertex next to it, which is a non-collider that is in $\bar{S}$. This means there is an edge from $\bar{S}$ to S, which is a contradiction with S being prefix. Since P has no colliders, it must satisfy $P \cap \texttt{Anc}[w] \cap \texttt{Anc}[w'] \neq \varnothing$. However, since $\texttt{Anc}[w] \cap \mathrm{src}(\bar{S}) \neq \texttt{Anc}[w'] \cap \mathrm{src}(\bar{S})$, $|\texttt{Anc}[w] \cap \mathrm{src}(\bar{S})| = 1$ and $|\texttt{Anc}[w'] \cap \mathrm{src}(\bar{S})| = 1$, $P \cap \texttt{Anc}[w] \cap \texttt{Anc}[w'] \neq \varnothing \in \bar{S}$. This means there is a non-collider on P that is in S, which would make it inactive given S, a contradiction. □

C.2. Proof of Lemma 4.7

To prove Lemma 4.7, we will make use of the following results.

Lemma C.2. *Let S be a prefix subset and $w \in \bar{S} \setminus (D_S \cup \mathrm{src}(\bar{S}))$. By Lemma 5.1, let $\texttt{Anc}[w] \cap \mathrm{src}(\bar{S}) = \{v\}$. Then for any $u \in S$, if there is no directed path from u to w that does not intersect $\mathrm{src}(\bar{S})$, then*

$$u \perp w \mid S \cup \{v\} .$$

Proof. Assume on the contrary that $u \not\perp w \mid S \cup \{v\}$. Let P be an active path from u to w conditioned on v. If $P \cap S \neq \{u\}$, then consider the last node on P that is in S. Since $w \notin S$, this node must be pointing into a node in $\bar{S}$ on P, which makes this node a non-collider. However, this node belongs to S, which means P is inactive given $S \cup \{v\}$, and thus $P \cap S = \{u\}$.

There is also no collider on P. Otherwise consider the first collider; it will be in $\bar{S}$ since $P \cap S = \{u\}$. However, since P is active, it will be in $\texttt{Anc}[S \cup \{v\}]$. This is impossible since $v \in \mathrm{src}(\bar{S})$ and from Lemma C.1, we know that $S \cup \{v\}$ is prefixed. Therefore, P must be a directed path from u to w, where the node x adjacent to u is in $\bar{S}$. This means that $x \in \texttt{Anc}[w]$. If $x \notin \mathrm{src}(\bar{S})$, then P is a directed path from u to v that does not intersect $\mathrm{src}(\bar{S})$. If $x \in \mathrm{src}(\bar{S})$, since $\texttt{Anc}[w] \cap \mathrm{src}(\bar{S}) = \{v\}$, then it must hold that $x = v$. This means P is inactive given $S \cup \{v\}$, a contradiction. □

Lemma C.3. *Let S be a prefix subset and $v \in \mathrm{src}(\bar{S})$. Then for any $w \in \bar{S}$, either $v \perp w \mid S$ or $w \in \texttt{Des}[v]$.*

Proof. Suppose $w \notin \texttt{Des}[v]$; we will show that $v \perp w \mid S$. Assume on the contrary that $v \not\perp w \mid S$. Let P be the active path between v and w given S. If P intersects with S, then consider the last vertex on P that is in S. This vertex must be a non-collider since $w \notin S$. This contradicts P being active given S. Therefore, P is fully in $\bar{S}$. Since $v \in \mathrm{src}(\bar{S})$, we have $P : v \to \ldots w$. Since $w \notin \texttt{Des}[v]$, there must be a collider on P. However, this collider is not in S and S is prefix, which means that P is active given S, a contradiction. □

We now prove Lemma 4.7, restated below.

Lemma 4.7. *Let S be a prefix vertex set. For $w \in \bar{S} \setminus D_S$, if $w \notin \mathrm{src}(\bar{S})$ and there is* no *covered edge from $\texttt{Anc}[w] \cap \mathrm{src}(\bar{S})$ to $\texttt{Anc}[w]$, then $w \in E_S \cup F_S$.*

Proof of Lemma 4.7. Assume $w \notin \mathrm{src}(\bar{S})$. For contradiction, assume that there is no covered edge from $\texttt{Anc}[w] \cap \mathrm{src}(\bar{S})$ to $\texttt{Anc}[w]$ and $w \in \bar{S} \setminus (D_S \cup E_S \cup F_S)$.

Since $w \notin \mathrm{src}(\bar{S})$, there is $v \in \mathrm{src}(\bar{S}) \cap \texttt{Anc}[w]$. As $v \in \texttt{Anc}[w]$, there is a directed path from v to w. Consider the longest directed path P from v to w. Let v' be the adjacent vertex to v on P, i.e., $P : v \to v' \cdots \to w$. By Lemma 4.5 and $w \notin D_S$, we must have $v' \notin D_S$. Since $v \to v'$ is not a covered edge, there could only be two cases:

- There is $u \to v$ such that $u \notin \texttt{Pa}(v')$. Since $v \in \mathrm{src}(\bar{S})$, we must have $u \in S$. Note that $u \not\perp v \mid S$ and $v \not\perp w \mid S$ and the second condition of the lemma is not met. We must have $u \not\perp w \mid S \cup \{v\}$. By Lemma C.2, there must exist a directed path Q from u to w that does not intersect $\mathrm{src}(\bar{S})$. Consider the path between u and w' by joining Q and the directed path from v' to w. Since this path does not intersect with $S \cup \mathrm{src}(\bar{S})$ and w is the only collider on it, we know that this path is active given $S \cup \{v\} \cup \{w\}$. Thus $u \not\perp v' \mid S \cup \{v\} \cup \{w\}$. Since the second condition of the lemma is not met, we must have $u \not\perp v' \mid S \cup \{v\}$. By Lemma C.2, there must exist a directed path from u to v' that does not intersect $\mathrm{src}(\bar{S})$. Since $u \notin \texttt{Pa}(v')$, this path has at least length two. Let k be the vertex on this path such that $k \to v'$. Note that $v \to v' \leftarrow k$. Therefore $v \not\perp k \mid S \cup \{v'\}$. Since $v' \notin D_S$, we must have $v \not\perp k \mid S$. By Lemma C.3, we must have $k \in \texttt{Des}[v]$. Thus we can increase the length of the directed path P by replacing $v \to v'$ with $v \to \cdots \to k \to v'$. This contradicts P being the longest directed path from v to w.

- There is $k \to v'$ such that $k \notin \texttt{Pa}(v)$. Note that $v \to v' \leftarrow k$. Therefore $v \not\perp k \mid S \cup \{v'\}$. Since $v' \notin D_S$, we must have $v \not\perp k \mid S$. By Lemma C.3, we must have $k \in \texttt{Des}[v]$. Thus we can increase the length of the directed path P by replacing $v \to v'$ with $v \to \cdots \to k \to v'$. This contradicts P being the longest directed path from v to w.

This completes the proof. □

C.3. Remarks

The above proofs suffice as intermediate results to show Theorem 3.3, which we proved in Section 5.

D. Missing Proofs of Interventions

D.1. Proof of Lemma 6.1

Lemma 6.1. *Let S be a prefixed vertex set. Given an intervention on vertices $I \subseteq V$. For any $u \notin I$, we have $u \not\perp_I v$ for some $v \in I \setminus S$ if and only if $u \in \texttt{Des}(I \setminus S) \setminus (I \setminus S)$.*

Proof. For each $v \in I$, denote $\texttt{Des}(v, I)$ as the set of $u \in I$ such that $u \not\perp_I v$. We first show that $\texttt{Des}(v, I)$ is equal to the set of descendants of v such that there exists a directed path from v which is not cut by I.

Let $u \notin I$ such that $u \perp_I v$. If $u \in \texttt{Des}(v, I)$, then there exists a directed path from v to u in the modified DAG $\mathcal{G}^I$, which means it is active given $\varnothing$, a contradiction. Therefore the only if direction is proven.

For the if direction, let $u \notin I$ such that $u \not\perp_I v$. Then there is an active path $P : u - \cdots - v$ in the modified DAG $\mathcal{G}^I$. Since P is active given $\varnothing$, there is no collider on P. Furthermore, since all incoming edges to any vertex in I are removed in the modified DAG, this path must be $P : u \leftarrow \cdots \leftarrow v$ and satisfies $P \cap I = \{v\}$. Thus $u \in \texttt{Des}(v, I)$.

Next we show that $\cup_{v \in I \setminus S} \texttt{Des}(v, I) = \texttt{Des}(I \setminus S) \setminus (I \setminus S)$. This will prove the lemma.

Since $\texttt{Des}(v, I)$ is equal to the set of descendants of v such that there exists a directed path from v which is not cut by I, it is clear that $\cup_{v \in I \setminus S} \texttt{Des}(v, I) \subseteq \texttt{Des}(I \setminus S) \setminus (I \setminus S)$. Now let $u \in \texttt{Des}(I \setminus S) \setminus (I \setminus S)$. Then there is $v \in I \setminus S$ such that $u \in \texttt{Des}(v)$. Consider the directed path from v to u and let v' be the last vertex on this path that is in I. Then $u \in \texttt{Des}(v', I)$. By the fact that S is prefix, we have from $v \notin S$ that $v' \notin S$. Thus $u \in \texttt{Des}(v', I) \subset \cup_{v \in I \setminus S} \texttt{Des}(v, I)$ and $\cup_{v \in I \setminus S} \texttt{Des}(v, I) \supseteq \texttt{Des}(I \setminus S) \setminus (I \setminus S)$. We therefore have $\cup_{v \in I \setminus S} \texttt{Des}(v, I) = \texttt{Des}(I \setminus S) \setminus (I \setminus S)$. □

D.2. Proof Lemma 6.2

Lemma 6.2. *Let S be a prefix subset. Given an intervention I and $v \in I \setminus S$, denote $H_S^I(v) = \{u \notin S \cup \texttt{Des}[I \setminus S] : u \not\perp v \mid V \setminus \texttt{Des}[I \setminus S]\}$.*[12] *Then $\texttt{Pa}(v) \setminus (S \cup \texttt{Des}[I \setminus S]) \subseteq H_S^I(v) \subseteq \texttt{Anc}(v) \setminus (S \cup \texttt{Des}[I \setminus S])$.*

Proof. On one hand, let $u \in \texttt{Pa}(v) \setminus (S \cup \texttt{Des}[I \setminus S])$. Clearly $u \notin S \cup \texttt{Des}[I \setminus S]$ and $u \not\perp v \mid V \setminus \texttt{Des}[I \setminus S]$. Thus $u \in H_S^I(v)$, which proves $\texttt{Pa}(v) \setminus (S \cup \texttt{Des}[I \setminus S]) \subseteq H_S^I(v)$.

On the other hand, let $u \in H_S^I(v)$. Since $u \not\perp v \mid V \setminus \texttt{Des}[I \setminus S]$, let $P : u - \cdots - v$ be the active path given $V \setminus \texttt{Des}[I \setminus S]$. Then all non-colliders on P are in $\texttt{Des}[I \setminus S]$, and all colliders on P are in $\texttt{Anc}[V \setminus \texttt{Des}[I \setminus S]]$

Note that $\texttt{Des}[I \setminus S] \cap \texttt{Anc}[V \setminus \texttt{Des}[I \setminus S]] = \varnothing$. Otherwise let $w \in \texttt{Des}[I \setminus S] \cap \texttt{Anc}[V \setminus \texttt{Des}[I \setminus S]]$. Since $w \in \texttt{Anc}[V \setminus \texttt{Des}[I \setminus S]]$, there is $w' \notin \texttt{Des}[I \setminus S]$ such that $w' \in \texttt{Des}[w]$. However $w \in \texttt{Des}[I \setminus S]$, which means $w' \in \texttt{Des}[I \setminus S]$. A contradiction.

If P has a collider $y \to x \leftarrow z$, then z is either a non-collider or v. Either case we have $z \in \texttt{Des}[I \setminus S]$. This means $x \in \texttt{Des}[I \setminus S]$. However, $x \in S \cup \texttt{Anc}[V \setminus \texttt{Des}[I \setminus S]]$ and $\texttt{Des}[I \setminus S] \cap \texttt{Anc}[V \setminus \texttt{Des}[I \setminus S]] = \varnothing$. A contradiction. Thus there is no collider on P.

Denote the vertex next to u on P as t, i.e., $P : u - t - \cdots - v$. Then t is either a non-collider or v. Either case $t \in \texttt{Des}[I \setminus S]$. Therefore it must be $u \to t$, otherwise $u \in \texttt{Des}[I \setminus S]$. Furthermore, as P has no colliders, it must be $P : u \to t \cdots \to v$. Thus $u \in \texttt{Anc}(v)$. Together with $u \notin S \cup \texttt{Des}[I \setminus S]$, we have $u \in \texttt{Anc}(v) \setminus (S \cup \texttt{Des}[I \setminus S])$. This

[12]Note that $\texttt{Des}[I \setminus S]$ can be obtained via Lemma 6.1 and $(\texttt{Des}(I \setminus S) \setminus (I \setminus S)) \cup (I \setminus S)$.

proves $H_S^I(v) \subseteq \texttt{Anc}(v) \setminus (S \cup \texttt{Des}[I \setminus S])$. □

D.3. Proof of Lemma 6.4

Lemma 6.4. *Let S be a prefix vertex set. Then $\forall w \in J_S^I$, we have $\texttt{Des}(w) \subseteq J_S^I$. Furthermore, $J_S^I \cap \mathrm{src}(\bar{S}) = \varnothing$.*

Proof. Assume on the contrary that $v \in J_S^I$ but $w \in \texttt{Des}(v)$ such $w \notin J_S^I$. Since $v \in J_S^I$, there could be two cases:

- $v \in \texttt{Des}(I \setminus S)$. Then clearly $w \in \texttt{Des}(v) \subseteq \texttt{Des}(I \setminus S)$. A contradiction.
- $v \in I \setminus S$ and there is $u \in H_S^I(v) \cap \bar{S}$. Then $w \in \texttt{Des}(v) \subset \texttt{Des}(I \setminus S)$. A contradiction.

Next we show that $\mathrm{src}(\bar{S}) \cap J_S^I \neq \varnothing$. Note that by $\mathrm{src}(\bar{S}) \cap \texttt{Des}(\bar{S}) = \varnothing$ and $\texttt{Des}(I \setminus S) \subseteq \texttt{Des}(\bar{S})$, we have $\mathrm{src}(\bar{S}) \cap \texttt{Des}(I \setminus S) = \varnothing$. In addition, for any $v \in \mathrm{src}(\bar{S}) \cap (I \setminus S)$, by Lemma 6.2, we have $H_S^I(v) \subset \texttt{Anc}(v) \setminus S = \varnothing$. Thus $\mathrm{src}(\bar{S}) \cap \{v \in I \setminus S : \ H_S^I(v) \cap \bar{S} \neq \varnothing\} = \varnothing$. We then have $\mathrm{src}(\bar{S}) \cap J_S^I = \varnothing$. □

D.4. Remarks

The above proofs suffice as intermediate results for proving Theorem 6.5. Then together with Lemma 6.6 (proven in Section 6.1), we can prove Theorem 3.4, which is given in Section 6.2.

E. Details of Numerical Experiments

Implementation Details. For `FCI` and `RFCI`, we used the implementations in Kalisch et al. (2024), which is written in R with C++ accelerations. For `PC` and `GSP`, we used the implementation in Squires, which us written in python. Our method, `CCPG`, is written in python. The acceleration of R (with C++) can be viewed by comparing two implementations of PC (stable) in Figure 6.

Remark on causal sufficiency. Among the constraint-based methods, we marked the ones that do not assume causal sufficiency in Figure 7. These methods run additional tests to check for unobserved causal variables, and therefore might require more samples compared to, e.g., `PC`, since the underlying system we test on satisfies causal sufficiency.